

REMARKS/ARGUMENTS

The arguments presented herein include the arguments Applicants discussed with the Examiner during phone interview dated March 25, 2011. The Examiner requested Applicants to submit the discussed arguments for reconsideration, which Applicants present herein. Applicants submit that the arguments presented herein make the substance of the phone interview of record to comply with 37 CFR 1.133. If the Examiner believes that further information on the interview needs to be made of record to comply with the requirements, Applicants request the Examiner to identify such further information.

Applicants submit that any amendment to the claims herein does not comprise acquiescence or admission that any canceled, amended or supplemented subject matter that existed prior to the amendments herein is not patentable. Applicants reserve the right to pursue claimed subject matter as presented prior to the amendments herein during subsequent prosecution of the present application and in any continuation or related applications.

Applicants amend claim 5 to correct a mistake to say the “method of claim 1”.

1. Amended Claims Comply with 35 U.S.C. §112, par. 2

The Examiner rejected claims 7-9, 11-13, 20-22, 24-26, 33-35, and 37-39 as indefinite (35 U.S.C. §112, par. 2). (FOA, pg. 6)

Applicants amend claims 7, 8, 20, 21, 33, and 34 to clarify the claim language of a local copy of the target shared file as disclosed in at least para. 25 of the filed Specification to overcome the indefiniteness issues raised by the Examiner.

Applicants amend claims 11, 24, and 37 to depend from claims 9, 22, 35, respectively, which provide antecedent basis for the elements of claim 11, 24, and 37.

Applicants amend claim 22 to correct a grammar error.

Applicants request the Examiner to enter these amendments after final to clarify the claim language and remove the Sec. 112 rejection.

Applicants submit that these amendments overcome the Sec. 112 rejection.

2. Claims 1-6, 14-19, and 27-32 are Patentable Over the Cited Art

The Examiner rejected claims 1-6, 14-19, and 27-32 as obvious (35 U.S.C. §103) over Patel (U.S. Patent Pub. No. 2002/0157089). Applicants traverse.

Claims 1, 14, and 27 recite installing a program with a base computer, having a local device, wherein the base computer is adapted to access to a shared folder accessible to the client computers over a network, wherein installing the program on the base computer adds shared application components including executable files for the program to the shared folder and adds local application components of the program to the local device used by the base computer, and wherein installing the program enables the base computer to run the program by accessing by accessing the shared application components and local application components of the program in the shared folder and in the local device; creating an image of the local device of the base computer including the local application components of the program; and providing the image to the client computers to apply the local application components of the program to local devices of the client computers, wherein applying the image to the local devices of the client computers enables the client computers to access the shared application components in the shared folder to run the program, wherein the clients execute the local application components and the shared application components to run the program.

The Examiner cites paras. 22, 128, 157, 159, 269, 459 and 460 of Patel as teaching the limitation of installing a program with a base computer, having a local device, wherein the base computer is adapted to access to a shared folder accessible to the client computers over a network, wherein installing the program on the base computer adds shared application components including executable files for the program to the shared folder and adds local application components of the program to the local device used by the base computer, and wherein installing the program enables the base computer to run the program by accessing the shared application components and local application components of the program in the shared folder and in the local device. (FOA, pgs. 3, 11-12) Applicants traverse.

The cited para. 22 mentions starting a streamed application program, and the cache manager requests application pages from the server not resident in an application page cache. The cited para. 128 mentions application file pages 111, which is an output of the builder and put on the application server 107 to serve the bits to the client. The cited para. 157 mentions an application install monitor 403 that monitors installation to figure out files and registry entries created and other changes during installation. The cited para. 159 mentions an app stream packager 404 that takes information gathered by the profiler and monitor and forms application file pages and the stream app install block 405. The cited para. 269 mentions that users have

identical copies of the application files, and distributing a single fixed image across all servers. The cited para. 459 mentions that the AppInstallBlock code is created by a builder program that monitors the installation process of a local version of application installation program and records changes to the system, including environmental variables added or removed from the system or files added or modified in system directories. Files added to the application directory are not recorded. The builder profiles the application to obtain critical pages needed to run application. The cited para. 460 mentions that the AppInstallBlock and runtime data are packaged into a Streamed Application Set (SAS) by the builder and uploaded to the application server. After the client is subscribed to an application and before the application runs, the AppInstallBlock is sent by the server to the client. The SAS client invokes the default and application specific initialization code. The default and application specific procedure from the server process the data to make the machine ready for streaming that application.

The cited Patel discusses how an installation is monitored to build a package of an installation that is then streamed to clients. The cited Patel does not disclose the claim requirements that the base computer add shared application components, including executable files for the program, to a shared folder, and adds local application components of the program to install to a local device of the base computer. Instead, the cited Patel discuss how a builder profiles an application and packages runtime data into an SAS to upload on the server for the client to stream. There is no disclosure of the claim requirements of how shared application components used to execute the program are added to a shared folder the clients also access to run the program when local application components are added to the local device of the base computer to be provided to add to the local devices of the client computers. For instance, the Examiner has not shown where Patel discloses that the server install local application components in a local device of the server and shared application components in a shared folder that the clients, to which the application is deployed, access to run the application, such that both the local and shared application components are executed to run the application.

The Examiner cited para. 460 as disclosing the claim requirements of providing the image to the client computers to apply the local application components of the program to local devices of the client computers, wherein applying the image to the local devices of the client computers enables the client computers to access the shared application components in the

shared folder to run the program, wherein the clients execute the local application components and the shared application components to run the program. (FOA, pg. 12) Applicants traverse.

As mentioned, the cited para. 460 discusses how to make a machine ready for streaming a particular application. This does not disclose that the image is provided to client computers to apply local application components of the program to local devices of the client computers, and that the client computers access also the shared application components in the shared folder, including executable files, to run the program.

In the Final Office Action, the Examiner recognized that Patel does not disclose the claim requirements that installing the program enables the base computer to run the program by accessing the shared application components and local application components in the shared folder and local device, respectively. (FOA, pg. 13) To address this shortcoming, the Examiner found that it would be obvious to modify Patel to enable to access and execute program components in the shared folder. The Examiner found that modification would be motivated by the need to monitor the installation and execution of the program in its intended environment. (FOA, pg. 13). Applicants traverse.

The cited Patel nowhere mentions any motivation that a base computer install shared application components in a shared folder accessed by both the base computer and clients receiving the image from the base computer to run the installed program. Instead, Patel discusses how an installation is monitored to build a package or image that is then sent to clients to use. The Examiner has not shown any part of Patel that provides any motivation to have shared application components accessed by the client computers and the base computer from a shared folder along with the local application components accessed from their local devices. For instance, the Examiner has not shown why the discussion in Patel of packaging installation changes in the SAS package provides motivation to have the base computer install shared application components in a shared folder that is accessed by the base computer and client computers in addition to local application components installed locally on the clients and base computer. Applicants submit that by packaging installation changes in an SAS package to stream to the clients, Patel teaches away from any such motivation to install shared application components in a shared folder for access by the base computer and clients to run the program.

Moreover, once the application in Patel is streamed to the clients, the clients run the application locally. The Examiner has not shown where Patel teaches that the clients after

receiving the streamed application later have to run shared application components on a shared folder to run the program, where the base computer had added shared application components including executable files to the shared folder separate from the base computer.

Accordingly, claims 1, 14, and 27 are patentable over the cited art because the cited Patel does not teach or suggest all the claim requirements.

Claims 2-6, 15-19, and 28-32 are patentable over the cited art because the base claims 1, 14, and 27, from which they depend, are patentable over the cited art, and because the additional requirements of these claims in combination with the base and any intervening claims provide further grounds of patentability over the cited art. Moreover, the following claims provide further grounds of patentability over the cited art.

Claims 5, 18, and 31 depend from claims 1, 14, and 27, respectively, and further require that the image includes a driver that when loaded into the client computers causes the client computers to perform: intercepting a write request to a requested shared file in the shared folder; generate a mapping of the shared file to a local copy of the shared file in the local devices of the client computers; and applying the write to the local copy of the shared file in the local devices.

The Examiner cited paras. 181, 182, and 258 of Patel as teaching the additional requirements of these claims. (FOA, pgs. 4 and 15). Applicants traverse.

The cited para. 181 mentions that the application streaming file system is implemented on the client using a file system driver and a helper application. The file system driver receives requests from the OS for files belonging to the streaming file system, which are standard file system requests. The file system driver converts path names into file IDs, and forwards the requests to the user mode application to handle. The cited para. 182 mentions that the user mode program manages the cache of application file contents on the local file system and contacts the streaming server for file contents not in local cache. For each read, the user mode process checks if the data is in cache to return from cache. To obtain the contents, the user mode process sends the file identifier along with an offset at which to read and the number of bytes, and the application server sends back the requested data. The cited para. 258 mentions that the application server only services read requests with writes being handled on the client itself in a copy-on-write manner. Para. 211 of Patel mentions that a copy-on-write file system allows some applications to write configuration or initialization files where they want without rewriting the application and disturbing a local customizations.

The cited para. 258's mention that writes are handled on the client does not disclose a driver loaded onto client computers that intercepts a write to a shared file in the shared folder, to which the base computer added the shared application components, generates a mapping of the shared file to a local copy in the local device of the client, and then applies the write to the local copy. Although the cited Patel mentions writes handled on the client so that the client can rewrite without disturbing local customizations, this does not disclose that writes to a shared file in a shared folder, which has the shared application components, including executable files, used by the client computers to run the programs, are directed to a mapped-to local copy on the local devices of the clients.

Further, the cited discussion of converting path names does not disclose the claim requirement of a driver loaded onto client computers that intercepts a write to a shared file in the shared folder, to which the base computer added the shared application components, generates a mapping of the shared file to a local copy in the local device of the client, and then applies the write to the local copy.

Accordingly, claims 5, 18, and 31 provide additional grounds of patentability over the cited art because the additional requirements of these claims are not disclosed in the cited Patel.

3. Amended Claims 7-9, 11-13, 20-22, 24-26, 33-35, and 37-39 are Patentable Over the Cited Art

The Examiner rejected claims 7-9, 11-13, 20-22, 24-26, 33-35, and 37-39 as anticipated (35 U.S.C. §102) by Patel (U.S. Patent Pub. No. 2002/0157089). Applicants traverse.

Amended independent claims 7, 20, and 33 enabling access to the shared folders accessible over the network, wherein a shared file directory structure comprises a file directory structure of the shared folders; intercepting a write directed to a target shared file comprising one of the shared files in one of the shared folders; determining whether there is a mapping of the target shared file in the shared folder to a local copy of the target shared file in a local folder in the local device; generating a mapping of a file directory structure including the target shared file to a local copy of the file directory structure including a local copy of the target shared file in the local device in response to determining that there is no mapping of the file directory structure of the target shared file to the local copy of the target shared file in the local device, wherein the local copy of the file directory structure provides a partial view of the shared file directory

structure if target shared files in the shared file directory structure have not been accessed; and applying the write to the local copy of the target shared file.

The Examiner cited para. 142 of Patel as disclosing the pre-amended “generating a mapping” limitation (FOA, pg. 7), which as amended recites generating a mapping of a file directory structure including the target shared file to a local copy of the file directory structure including a local copy of the target shared file in the local device in response to determining that there is no mapping of the file directory structure of the target shared file to the local copy of the target shared file in the local device, wherein the local copy of the file directory structure provides a partial view of the shared file directory structure if target shared files in the shared file directory structure have not been accessed. Applicants traverse.

The cited para. 142 mentions a file spoof database at the client of a list of files the requests to which need to be redirected to the client streaming file system at the client. The client streaming file system serves all file system requests made by the application running on the client, and that reads and writes to files lead to page faults. A client cache manager is asked for the file and if it exists in the client cache, the client cache manager will get from there or forward to the application server if not local. See, para. 135 of Patel.

Patel discusses how files requested by the application are maintained locally at the client to use. The cited Patel does not disclose the claim requirements of building locally the file directory structure including a requested target shared file, wherein the local copy of the file directory structure provides a partial view of the shared file directory structure if target shared files in the shared file directory structure have not been accessed.

In the Response to Arguments, the Examiner further cited para. 182 of Patel in response to the above arguments. (FOA, pg. 4) Applicants traverse.

The cited para. 182 mentions that the user mode program manages the cache of application file contents on the local file system and contacts the streaming server for file contents not in local cache. For each read, the user mode process checks if the data is in cache to return from cache. To obtain the contents, the user mode process sends the file identifier along with an offset at which to read and the number of bytes, and the application server sends back the requested data.

The cited para. 182 does not disclose the claim requirements of building locally the file directory structure including a requested target shared file, wherein the local copy of the file

directory structure provides a partial view of the shared file directory structure if the target shared files in the shared file directory structure have not been accessed. The cited para. 182 discusses how files are cached locally, but does not disclose building locally a local copy of the file directory structure including a local copy of the target shared file, where the local copy of the file directory structure provides a partial view of the shared file directory structure. The Examiner has not shown where Patel discloses building a partial view of the file directory structure of the shared folders in a local copy of a file directory structure.

Accordingly, amended claims 7, 20, and 33 are patentable over the cited art because the cited Patel does not disclose all these claim requirements.

Claims 8-13, 21-26, and 34-39 are patentable over the cited art because the base claims 7, 20, and 33, from which they depend, are patentable over the cited art, and because the additional requirements of these claims in combination with the base and any intervening claims provide further grounds of patentability over the cited art. Moreover, the following claims provide further grounds of patentability over the cited art.

Conclusion

For all the above reasons, Applicant submits that the pending claims 1-9, 11-22, 24-36, 38, and 39 are patentable. Should any additional fees be required beyond those paid, please charge Deposit Account No. 50-0585.

The attorney of record invites the Examiner to contact him at (310) 553-7977 if the Examiner believes such contact would advance the prosecution of the case.

Dated: April 5, 2011

By: /David Victor/

David W. Victor
Registration No. 39,867